A Mobile Real-Time Data Collection and Analysis System for Farm Management

A Bachelors of Science Honors Thesis
Presented in Partial Fulfillment of the Requirements for
Graduation with Distinctions from the
Department of Computer Science and Engineering
At The Ohio State University

By

Nicholas David Larusso

****

The Ohio State University
May 2006

Honors Thesis Committee:
Assistant Professor Rajiv Ramnath, Advisor
Assistant Professor Stephen Boyles

Abstract

Food and agriculture is Ohio's most important industry and at the heart of that industry is the business of cattle farming. While the size of this industry in Ohio is large, a significant percentage of the farms are small family-owned and run. These farms typically hover closely around the line that separates profitable farms from those that lose money. Using principles from the Balanced Scorecard, a management technique new to the farming industry, and the appropriate use of technology, we seek to assist farmers in making better day-to-day management decisions for their farms with the expectation that this micro-level approach to managing will tip the balance towards profitability.

Inherent to management is the process of data collection. Farmers need to collect data while they work, and to do that, data collection software must be intuitive, user friendly, and fit into the process of the workflow. Further, farming is a technology-unfriendly environment. In response to these requirements, we have built a Personal Data Assistant (PDA)-based mobile, handheld application specifically designed for farm management. This technology will facilitate the process of collecting and analyzing data for farmers. Consequently, they will be able to make more accurate management decisions about the farm in real-time as they work. The application has been designed for easy use and is unobtrusively integrated into the farmer's environment and workflow.

In addition to considering usability and human-computing interface issues with such a management application, the overall architecture, was also considered. Recognizing that management is a common function across the domain of small farms, the objective was to design an extensible architecture that would lend itself to building micro-management applications for other farm operations.

Acknowledgements

To my advisor Rajiv Ramnath for teaching me about the research process and preparing me for further studies in the field of computer science.

To Stephen Boyles and Jeff McCutcheon for teaching me everything I know about cow-calf operations and giving me invaluable feedback during the project.

To my parents Dr. Daniel and Susan Larusso for always believing in me and offering an encouraging word or two when I needed it the most. Additionally, to my sisters and brothers-in-law for believing in me and showing support during all my pursuits even when my own self-confidence wavered.

To my wonderful girlfriend Imogen Pryce for being patient with me during those long days and nights and the sleep deprived grumpiness that followed.

Lastly to all the Computer Science and Engineering faculty and staff at The Ohio State University for giving me the solid educational foundation to complete this project and preparing me for the future.

Table of Contents:

Index of Figures:

I. INTRODUCTION

As a rule, software systems do not work well until they have been used, and have failed repeatedly, in real applications.
-Dave Parnas

Beef production is an enormous business in Ohio. There are currently 27,000 farms with cattle in Ohio ranking them 12th in the nation for the number of beef operations, with a total economic value of almost $1 billion[1]. Therefore, given their economic contribution, management techniques that improve their profitability could significantly impact Ohio's economy.

Much of the beef produced goes through the same process which begins with the cow-calf operations. A cow-calf operation is a farm consisting of a set of cows, and usually at least one bull with the purpose of breeding calves. The calves are usually raised on the farm until they are weaned, at which point they are sold to a feedlot. They remain at the feedlot where they are put on a special diet in order to quickly increase their weight. After the calves are grown to market weight, they are sold for slaughtering.

Small and medium sized cow-calf operations are in abundance in the Midwest and account for a significant portion of the production of natural beef, grown without added hormones. Due to lack of economies of scale[2] in small cow-calf operations, it is very difficult for the smaller farms to be consistently profitable [Miller et al. 2001].

[1] As reported by the Ohio Beef Council.

[2] Economies of Scale refers to the decreased per unit cost as output increases; as defined on Wikipedia.org.

Problem Statement

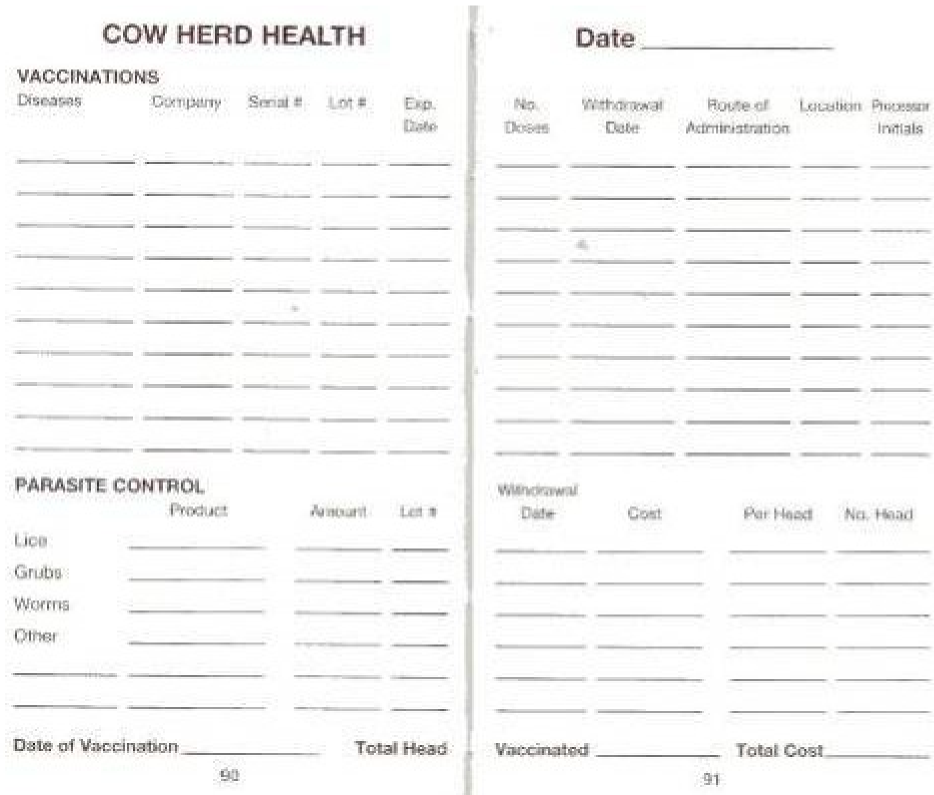
Small and medium sized cow-calf operations are in abundance in the Midwest and account for a significant portion of the production of natural beef, grown without added hormones. Due to the lack of economies of scale[2] in smaller cow-calf operations, it is very difficult for the smaller farms to be consistently profitable [Miller et al. 2001]. These farms are dependent on every single animal they produce, hence the need to carefully micromanage each one individually to ensure an efficient operation. Manual micromanagement, however, is difficult and time consuming because it requires accurate data to be collected on each animal. The goal of this project is to develop an inexpensive system using available technology, that would aid farmers in collecting and analyzing data in real-time in order to improve the farm's profitability by creating an extensible mobile application framework to extend its usefulness to other areas in the farm management domain.

This project was initiated as the result of the National Animal Identification System (NAIS) is program being put into action by the USDA. The NAIS requires that all animals be issued a unique animal identification number (AIN) or a group identification number (GIN) depending on how the specific type of animals are managed. The identification number would be used by the USDA to track each animal's whereabouts. Farmers will be required to assign each animal an identification number and report any location or ownership changes. The USDA's goal is that health officials will be able to use this information to locate possibly contaminated animals and locations within 48 hours of a disease's discovery. Currently the program is purely voluntary, but the USDA has plans to make it mandatory by the year 2008.

Because the implementation of this system would help prevent any kind of disease outbreak, it is likely that it will be implemented in the near future. This system will require farmers to collect data and track events for each of their livestock. Because data collection will be mandatory, it would benefit farmers to collect more data to help make managerial decisions about their farm. This system ties the AIN together with easy data collection and quick useful real-time analysis.

Background

Farm economists have recommended that farmers adopt new management techniques that require data collection and analysis. Most farmers do not collect data consistently, and of those farmers who do collect data, oftentimes the data is never reviewed, arguably because data collection is seen to be of little value or the process is too difficult and time consuming. Our biggest challenge, therefore, was to develop a system that offered useful analysis tools and was easy to use.



COW HERD HEALTH Date ____________

VACCINATIONS

Diseases	Company	Serial #	Lot #	Exp. Date	No. Doses	Withdrawal Date	Route of Administration	Location	Processor Initials

PARASITE CONTROL

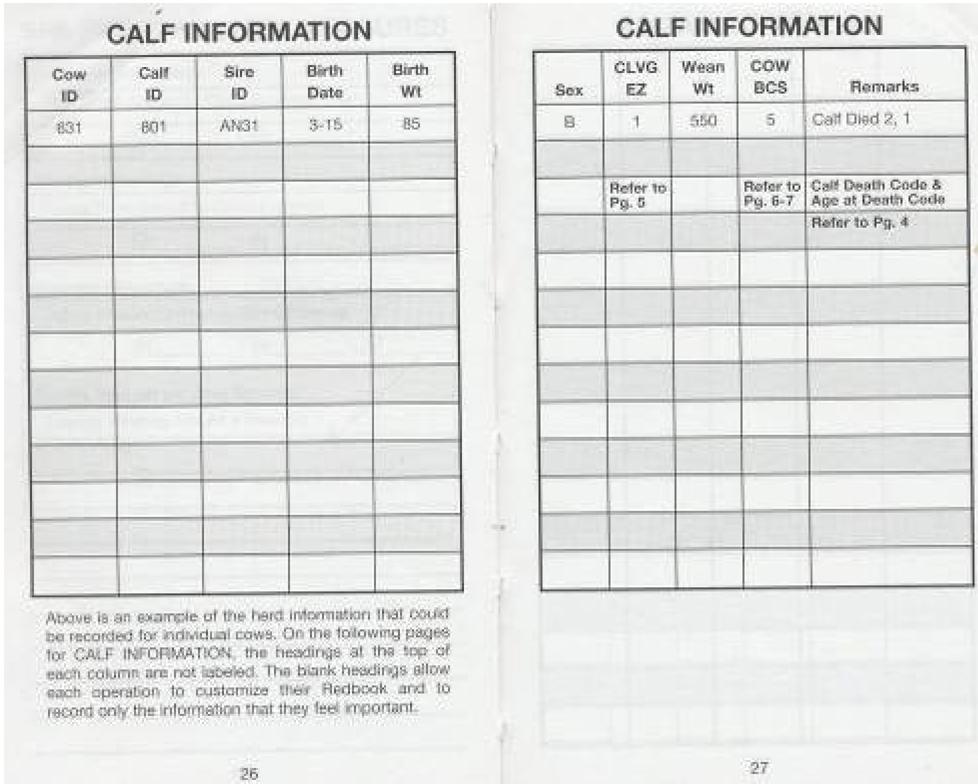
	Product	Amount	Lot #	Withdrawal Date	Cost	Per Head	No. Head
Lice							
Grubs							
Worms							
Other							

Date of Vaccination ____________ Total Head Vaccinated ____________ Total Cost ____________

90 91

Figure 1: IRM Livestock Medication Information

Several ‘paper and pencil’ methods for management exist, such as the Integrated Resource Management (IRM) book [IRM]. The IRM provides farmers with a plethora of information. It contains quick reference guides and includes multiple data-capture tables where the farmer can enter information on pastures, calf health and body condition scoring, calving and breeding activity, and other miscellaneous notes. The IRM allows farmers to capture information about livestock, feed, medication, and environmental conditions as they go about their work.



CALF INFORMATION

Cow ID	Calf ID	Sire ID	Birth Date	Birth Wt
831	801	AN31	3-15	85

Above is an example of the herd information that could be recorded for individual cows. On the following pages for CALF INFORMATION, the headings at the top of each column are not labeled. The blank headings allow each operation to customize their Redbook and to record only the information that they feel important.

26

CALF INFORMATION

Sex	CLVG EZ	Wean Wt	COW BCS	Remarks
B	1	550	5	Calf Died 2, 1
	Refer to Pg. 5		Refer to Pg. 6-7	Calf Death Code & Age at Death Code Refer to Pg. 4

27

Figure 2: IRM Calf Information

Paper and pencil methods permit freedom in terms of what data can be collected and its format. However, the problem with every 'paper and pencil' method is that data stored in a non-digital format makes analysis time consuming and error prone because it must be done manually. This means that farmers are less likely to do any analysis.

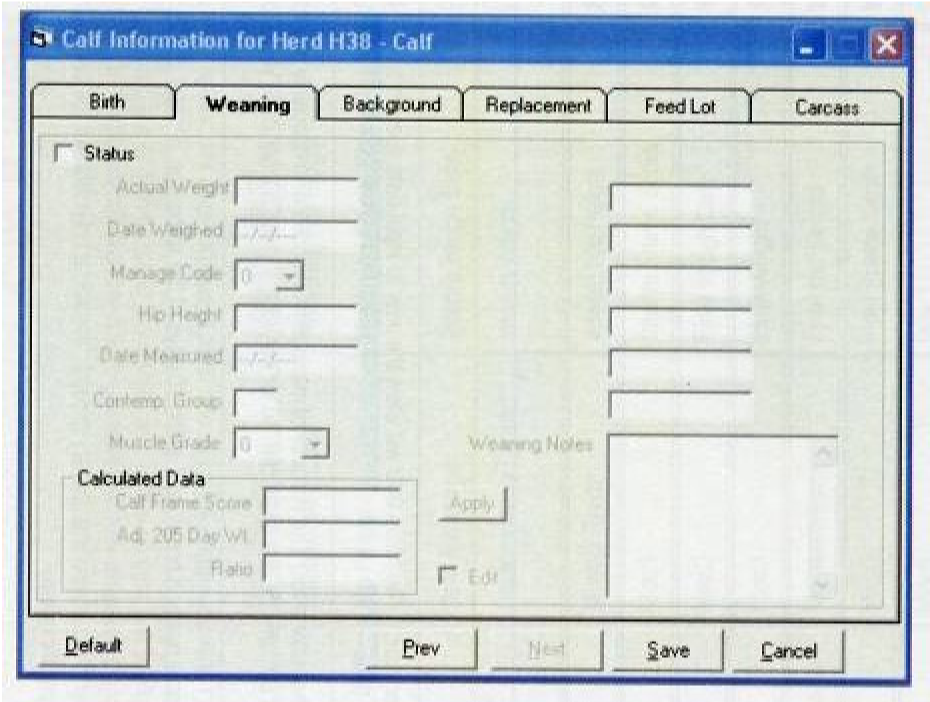


Figure 3: CHAPS Data Entry Software Screenshot

In an attempt to address this problem, a variety of software applications have been created specifically for farm management. One example is the Cow Herd Appraisal Performance Software (CHAPS). CHAPS is meant to be used for post-processing only, that is, after the farmer has collected the necessary data it must be entered into the computer and analyzed. The reports and information analysis given by this software are very informative, however, the process of collecting, entering, and then analyzing data is still time consuming. Also, CHAPS does not support the farmer's need to reference useful information while working in the field. Results from the analysis must be recorded outside the system in order to be useful to the farmer when he returns to his animals.

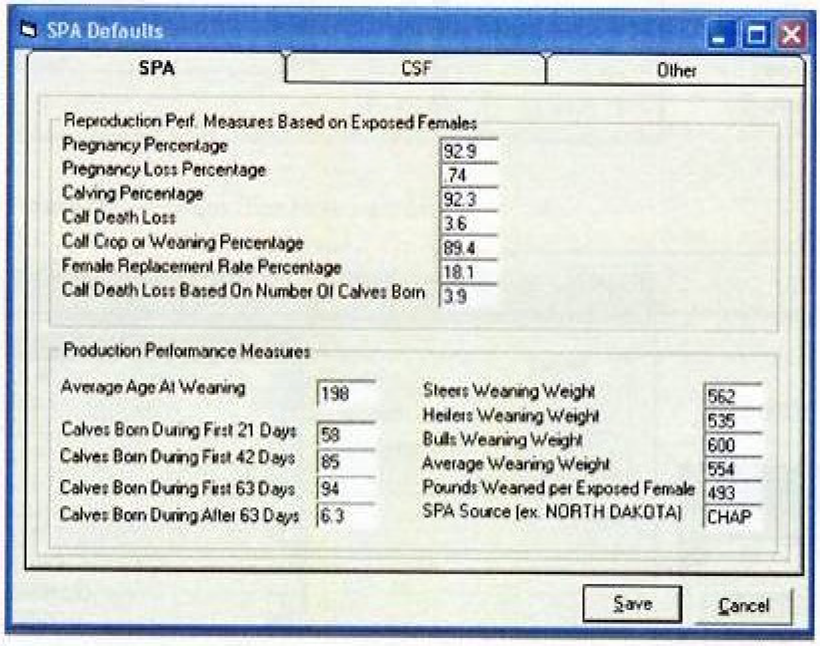


Figure 4: CHAPS Data Analysis Software Screenshot

A limited number of farm management systems include a companion system that uses mobile technology so users may access and enter data in the field. These systems offer the advantage of comprehensive data collection and analysis because they were built specifically for cattle management. However, these systems usually come with a significant price tag and offer a steep learning curve. Also, because the mobile portion of the system is dependent upon the software running on the PC, users are limited by the specific analysis capabilities of this software. Additionally, each program has its own specialized format for the data which adds even more restrictions to the software package.

In order to allow farmers to collect and analyze data in real time, a system needs to be mobile, small enough to be carried out onto the farm, usable by a single individual while working on the farm and relatively inexpensive. A personal data assistant (PDA) fits this profile extremely well. PDAs offer limited computing power and screen size, but are extremely mobile and relatively cheap (with a price range from $150 to $500). Many affordable PDA models now offer 64Mb or more of memory, enough to hold all of a user's contacts, calendar schedules, and to-do lists, in addition to running interactive applications with sufficient performance.

Lastly, there are benefits to extending the use of technologies to unconventional domains. Because farming has only recently begun adopting new technologies, there are relatively few options for farmers looking to add tools, such as automated management systems, to their farms. Because each system is only one solution to the problem, the first few are usually inadequate because they have no previous version to criticize. With more solutions available, it is more likely users will be able to find one that fits their specific needs.

II. REQUIREMENTS

Data Attributes

Since the computer science researcher had no previous experience in farming, much of the time spent building this application was in gathering requirements and understanding the domain. The first objective was to discover the attributes of the farming process that were vital to the management of the farm, and how to track them. Since our goal was to build a lightweight system with a focus on real-time data analysis instead of a data intensive system, only choosing elements that had the most bearing on whether a farm was profitable or not was an important step.

During the requirements gathering process, we ran into many of the "essential difficulties" described by [Faulk 1997]. Because the farming community is still adapting to the wealth of available technologies and the developer was not familiar with the specifics of farming, there was a vocabulary barrier to overcome. To resolve this issue, we performed a short ethnographic study by taking a visit to a local farm and learning the processes and needs of this farm. The farm was a small family operation composed entirely of two people. This cow-calf operation was unique in that a PDA was currently being used for data collection. Since no specialized software was used, the farmer was taking notes on the animals and feed resources on the standard memo application and analysis of the data was still a difficult task. Seeing how the PDA was used in the field, and what data were collected, gave us a better understanding of the system requirements.

We were able to determine that the major costs for farms are the feed, medication, and land[3]. Feed costs are reportedly the most critical of all the factors, and can account for more than 60% of the total cow-calf budget [Mangione]. Medication costs are also significant and tracking a livestock's medical record can be a useful tool for disease outbreak control and vaccination management as well. Since both feed and medication are perishable goods that are usually purchased and stored until use, we also had to factor in material waste and how it affects profitability. Other costs exist, but account for a significantly smaller portion of a cow-calf operation's budget. These costs may include things like marketing, fly tags, and fuel costs [Hughes et al. 1989].

From the work done in [Miller et al. 2001], it was reported that "total feed costs, followed by selling price of calves, and number of cows in the herd were the three most important factors explaining variation in profit." Better management of this single aspect has so much weight on the overall profitability of the farm; which is why so much focus was given to tracking the uses, wastes, and purchases of feed.

[3] Due to the time constraints, we decided that the land management portion was outside the scope of this project.

The second biggest factor in profitability is the selling price of calves, so we determined the factors that had a significant affect on this value. We were able to identify that the differentiating factors for calves are (a) their weight at the time of sale and (b)body condition scoring (BCS). The importance of the animal's weight is clear. The BCS is an identifier of the quality of the beef that animal will provide. It is a visually captured measure with a numerical value between 1 and 9, that describes the "degree of fatness of a cow" [Mangione]. Other measures may affect the selling price as well, but we chose to focus on these because both of these measures have a high correlation to the feed.

In addition to the factors that determine the profitability of a calf, it is important to understand the attributes of cows and bulls that create effective producers. Here the key attributes are (a) calving ease, (b) BCS, and (c) birthing trends. Calving ease is a measure of how easily an animal gave birth. This influences the amount of special attention a farmer must spend on a cow when she is birthing. The BCS of the cow during pregnancy is a critical measurement because it aids the farmer in managing the nutritional needs of the cow which in turn, enhances the cow's reproductive performance [Mangione]. Since this is a decisive time for both the producing cow and its offspring, this measurement can have an enormous effect on the profitability of a specific cow over time. The birthing trends of a cow will allow the farmer to make managerial decisions as to whether or not to sell a cow or bull if that animal is not producing consistently. Tracking offspring also allows for loose tracking of lineage which is also an important factor when performing in-depth analysis of a livestock because it is possible to trace the sale price of a cow's offspring to tell if a cow-calf pair has made or lost money over the year. This also allows users to view trends of quality. For instance, if a cow is consistently producing offspring that have high selling prices, it is a good producer.

The analysis we sought to offer is similar to what would be available to someone using the IRM. Given the data this system aims to collect, the analysis is focused on whether or not an animal is covering its costs by producing calves that have a high selling price. To calculate this, we figure out the total cost of the cow and calf until it is sold and subtract that amount from the selling price to get the animal's profit. Additionally, we also provide the ability for viewing trends in values like BCS, calves produced, and calving ease. Possibly the system's most useful analysis tools is simply the constant organization of the data as it is collected, allowing the user to focus on understanding what the data means.

III. SYSTEM ARCHITECTURE

Hardware

As mentioned earlier, one of the goals of this system was for the technology to be integrated into the workflow of the farm which means that the farmer should be able to enter data seamlessly throughout his workday. PDAs can be easily carried in the field and used with minimal interaction from the user and are much more affordable and easier to use than a Tablet PC or a laptop.

Our PDA is an HP iPAQ running the Pocket PC operating system. This specific PDA model was chosen because of its availability and ease of development. The Pocket PC operating system offers developers a complex set of tools, such as XML parsing and better exception management, through the built-in .Net Compact Framework (CF). In addition, through the Visual Studio Integrated Development Environment (IDE), creating a Pocket PC application is similar to developing a Windows application.

There are negative aspects to basing the development on this system as well. Because Pocket PC is a complex operating system, battery life of the device is reduced. Additionally, because this system was built using the .Net CF, the software is less portable. However, given the short development time, these negative aspects were outweighed by the shallow learning curve and available development tools.

Implementation Language

When deciding upon an implementation language, the main features we looked for were ease of user interface creation, object orientation capabilities, and a shallow learning curve. C# offered a simple syntax and the language has automatic garbage collection. C# is one of the languages in the Visual Studios Integrated Development Environment (IDE). This IDE offered additional tools, such as visual debugging, a Pocket PC emulator, and a 'drag and drop' method of creating user forms. These tools and features made C# and the Visual Studio's IDE a natural choice for development.

Data Storage

Because much of the work done by this system was concerned with tracking data, the method of storage was an important factor. We had many concerns when deciding how to store this data: memory, reading and writing speed, transferability, extensibility, and cost. We found a natural fit in XML because it was the mechanism that best addressed these concerns.

IV. SYSTEM DESIGN

Application Architecture and Software Development Methodology

For this application, we wanted to be able to make the software fit a model that could be used across the animal farming domain, not just in cow-calf operations. Thus, an aim for this project was to construct an architecture that could be fit to other businesses in the domain, such as chicken or sheep operations, with minor changes. Because of this criterion, it was especially important to take full advantage of encapsulation and data hiding.

Our approach was to first understand the requirements needed for a cow-calf operation as explained earlier. Next, we built a cause and effect diagram to clarify the sources of expenses and revenue and how they fit together; this helped us break the components down to create classes for the application.

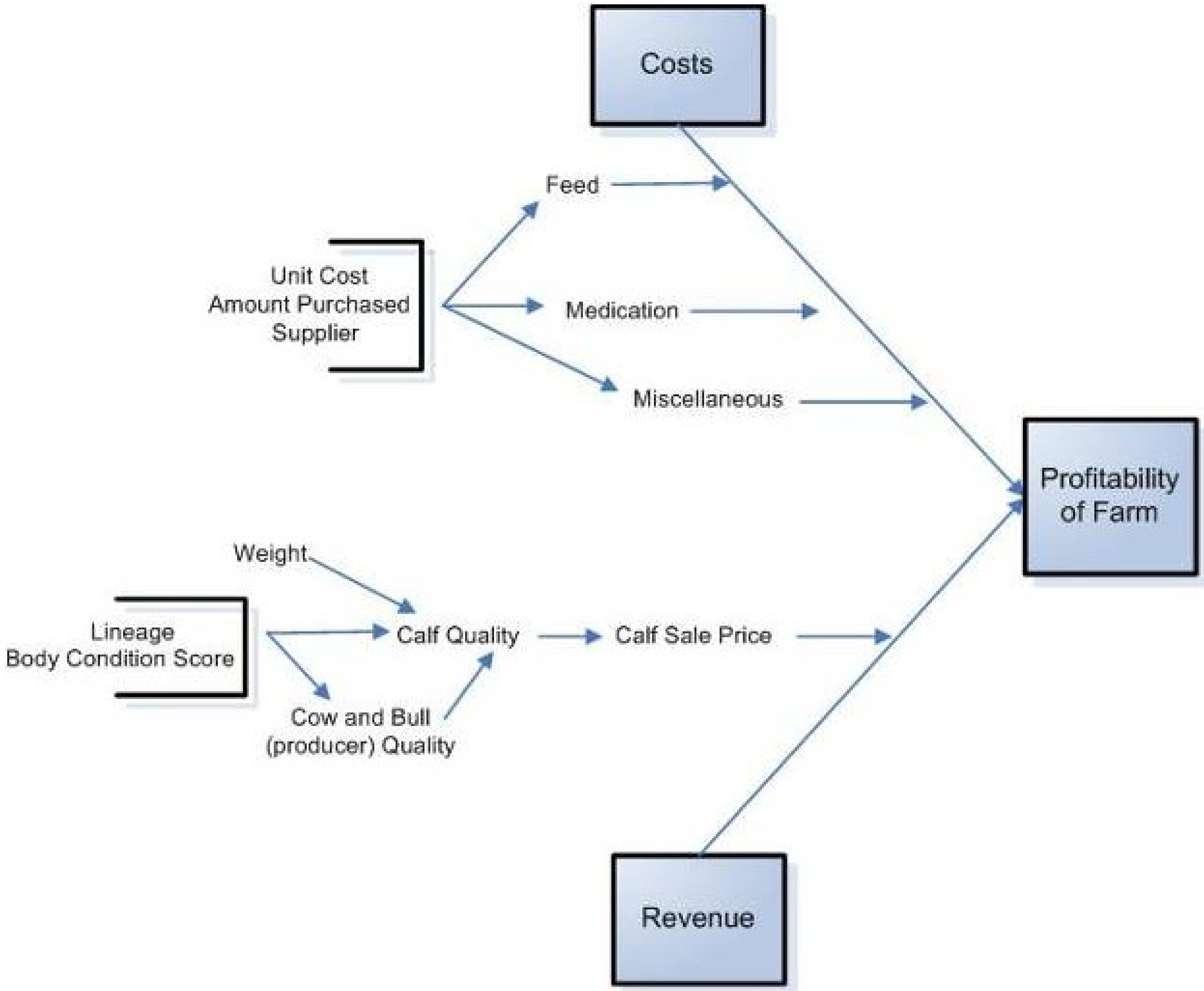


Figure 5: Farming Cause and Effect Diagram

This diagram visually exemplifies the need to carefully manage the multiple costs while maximizing the product selling price. Here we can also see important points of the animals that should be tracked in order to improve quality which in turn improves the sale price of the calf.

System Class Model

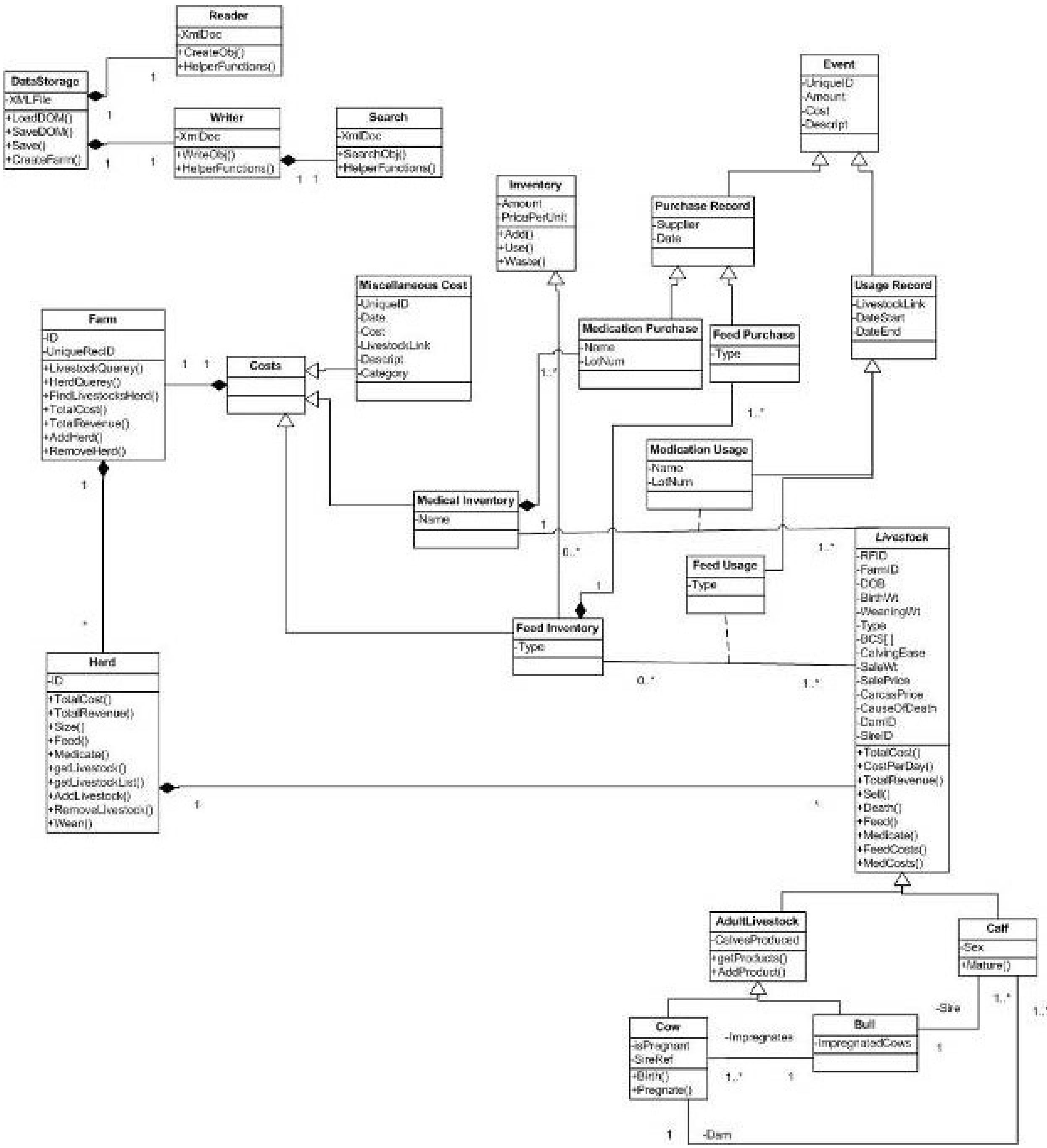


Figure 6: System Class Model

The above class model displays the different domain classes, their attributes and methods, and how they interact. Below, the class diagram is broken up into smaller more manageable groups for discussion. Also shown above are the implementation classes for reading and writing data.

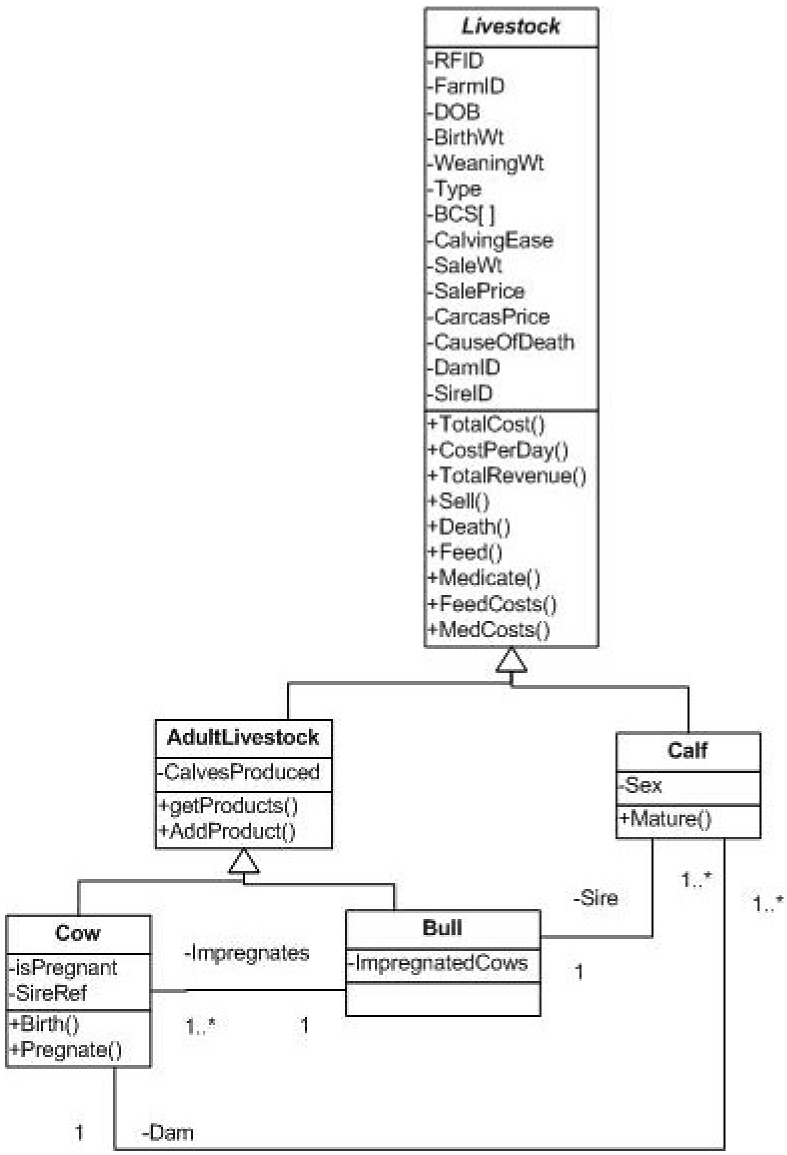


Figure 7: Livestock Class

Livestock

The attributes of the Livestock class are those relevant for all animals. The selection of these specific attributes is described in the requirements section. The methods are a collection of events that livestock may participate in, such as feeding and medicating, as well as management analysis points, such as cost per day, to help farmers judge whether or not the livestock is performing well. Inheriting from the livestock class are classes describing the product, Calf, and the producers, AdultLivestock.

Most of the time, calves remain on a farm for less than a year until weaning and are sold soon after that, additional information for these animals is not necessarily required. In the case that a farmer wishes to keep a calf as a replacement for a bull or cow, the only information necessary to make this transition is the gender of the animal. Likewise, the only method specifically required for the Calf is the ability to make the transition from Calf to either Cow or Bull.

The AdultLivestock's only attribute is a list of calves produced by the animal. Keeping track of the calves produced is a good metric for production consistency. This data also gives the farmer a link from producer to product so the quality of the calf can be evaluated and will be reflected on the cow and bull that created it.

The Cow and Bull classes simply add functionality specific to the production, of calves. The Cow class contains a method to give birth, which creates a new Calf and sets the Cow to be its mother, or dam. Likewise, the Bull class contains a method to impregnate a Cow, which identifies the Bull as the father, or sire, of the next Calf born. This information enables users to diagnose production issues by comparing the pregnancy and birthing histories of the cows and bulls. If a single cow did not get pregnant one year, she is accountable, but if multiple cows all bred with a single bull did not produce, then the bull is accountable.

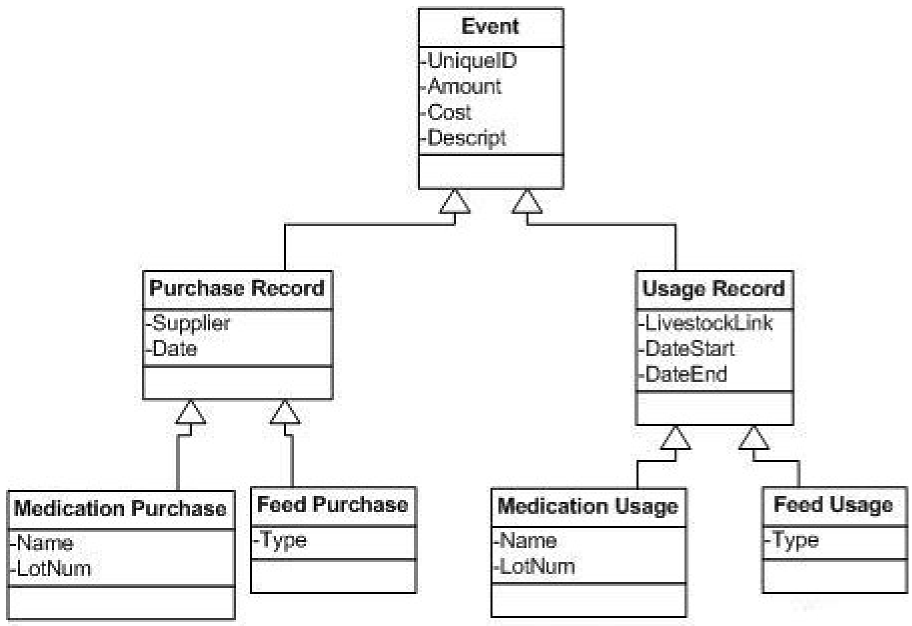


Figure 8: Event Class Hierarchy

Event

The event class hierarchy keeps track of the purchases, uses and wastes of feed and medication. A purchase occurs on a specific date and is linked to a supplier. Whereas a usage occurs over a range of time and is linked to a specific animal. Purchases help the farmer keep track of how money is being spent, while the usage records assign those costs to individual animals.

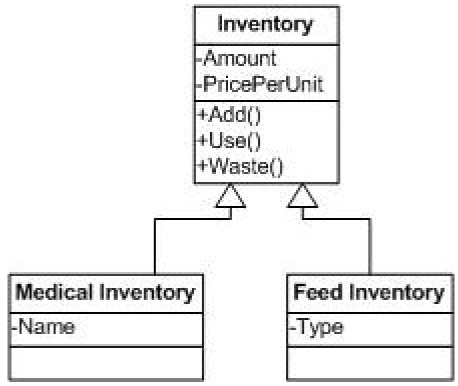


Figure 9: Inventory Class Hierarchy

Inventory

The inventory class hierarchy represents the farms medical and feed resources by keeping track of all assets on hand. Because feed and medication are both perishable, it is important to keep enough on hand while minimizing the amounts that are wasted. For both feed and medication, a separate inventory exists for each type of the product. For example, feed will contain separate inventories for gain, mineral, and hay. In addition to keeping track of the amounts on-hand, the Inventory class also serves as containers for each purchase, usage, and waste record. Keeping track of each record will become important when we look at the future work for this application, specifically the ability to create data mining tools to find trends over time.

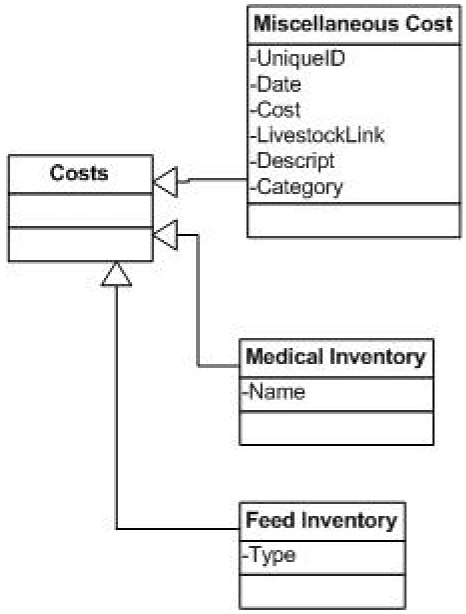


Figure 10: Costs class

Costs

The costs class aggregates all of the costs for the farm. This class is composed of the feed and medical inventories as well as the miscellaneous costs that can be associated with specific livestock (or the entire farm). Splitting the costs out from the Livestock object offers a more flexible organization since the type of costs that are collected are kept independent from the specific animal. This also allows for quick analysis at different levels of management. Because there is an association between the costs and individual animals, a user can look at how much the maintenance of one animal has cost over the past year. In addition, it may be interesting to see how expensive a specific herd has been, or at a higher level, the entire farm. Additionally, this class provides an abstraction between the individual groups of costs and the farm, making it easier to make changes to the individual costs the application can track.

The Miscellaneous Cost class offers the user some freedom in the information he or she can enter. Because it would not be feasible to create a separate class to manage every type of cost, this class is meant to cover everything other than feed and medication. Additionally, a category attribute was included in this class to allow a user a structure for organizing these costs.

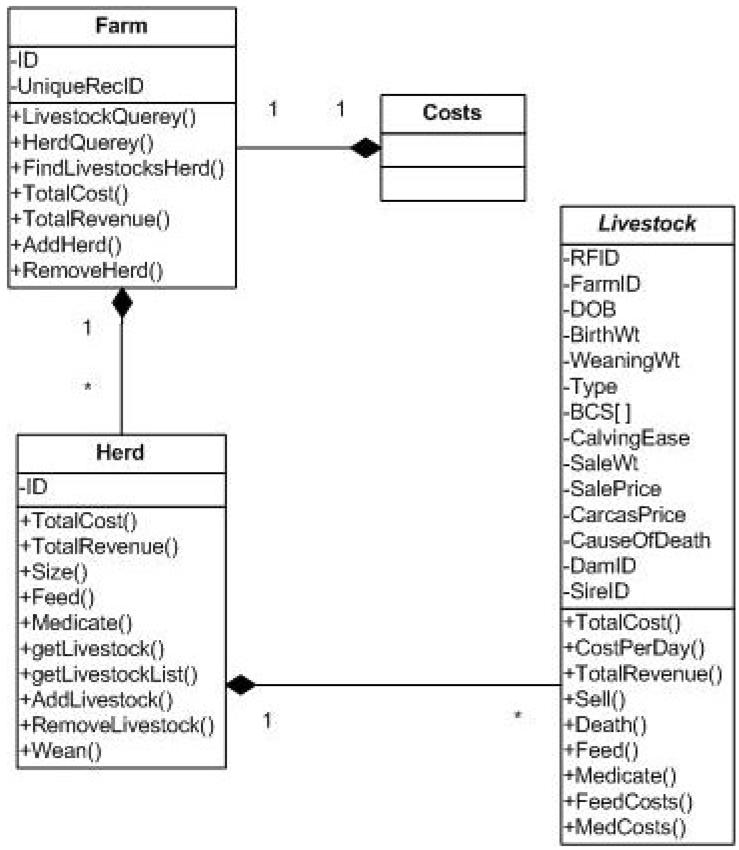


Figure 11: Farm and Herd Classes

Farm and Herd

The purpose of the herd class is two-fold. In addition to helping farmers keep track of how the cows and calves are actually grouped on their farm, it is also a way of pruning Livestock objects from the current memory context. In other words, a Herd is a method of memory management. Because PDAs have limited memory and computing capabilities, narrowing the scope to force a user to deal with one herd at a time ensures the PDA will have sufficient computational resources to handle the amount of data. If the user was able to view and manipulate every livestock in the farm from any point, the application would probably slow down significantly for larger farms because of memory constraints. This is the main reason that every livestock must belong to a herd, even it if is a herd containing a single animal.

Methods of the herd class are similar to those of the Livestock class. This is because herds offer a farmer the opportunity to perform group events such as feeding or medicating. The other methods, as in the Livestock class, are for management and analysis of groups of animals, such as calculating the feed costs, or the annual revenue.

The Farm class serves a purpose similar to the Herd class, to group a set of animals together. Like the Herd class, the Farm offers similar methods as well as general operations for adding and removing herds. This class also limits the focus of the application to a subset of the total data for memory management.

Costs, events, and other functions are maintained only in the Livestock class. When information from these attributes is requested by the containing classes, like Herd and Farm, their values are passed-up on demand. This way, if there is a change to the calculations, they are encapsulated in the Livestock class.

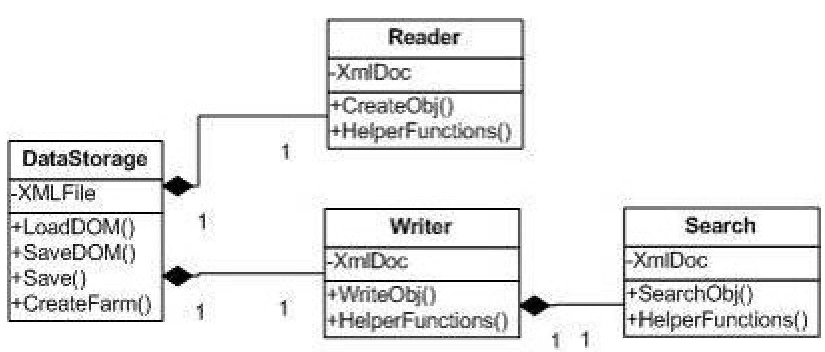


Figure 12: DataStorage Class

DataStorage

The DataStorage class and its subclasses encapsulate the input and output to an XML file. Any changes to the format of the file are thus contained within the Reader and Writer classes.

User Interface Design

Workflow Integration

Our goal for the user interface is to be able to replace a farmer's notebook with this system and have him or her be able to go about a regular workday, entering information when necessary, and not notice the difference much like the work completed by [Schraefel et al. 2004]. Our system, of course, offers data analysis capabilities not found on a normal notebook.

To decide how to present the data to the user, we focused on the key events in farming that require data collection. There are two major components to this application: livestock management and resource management. Livestock management includes animal transitions such as the birth of a new calf or the maturation of a calf to an adult, and operations like feeding and medicating. The possible transitions of a livestock's lifecycle are shown in the object's state diagram below.

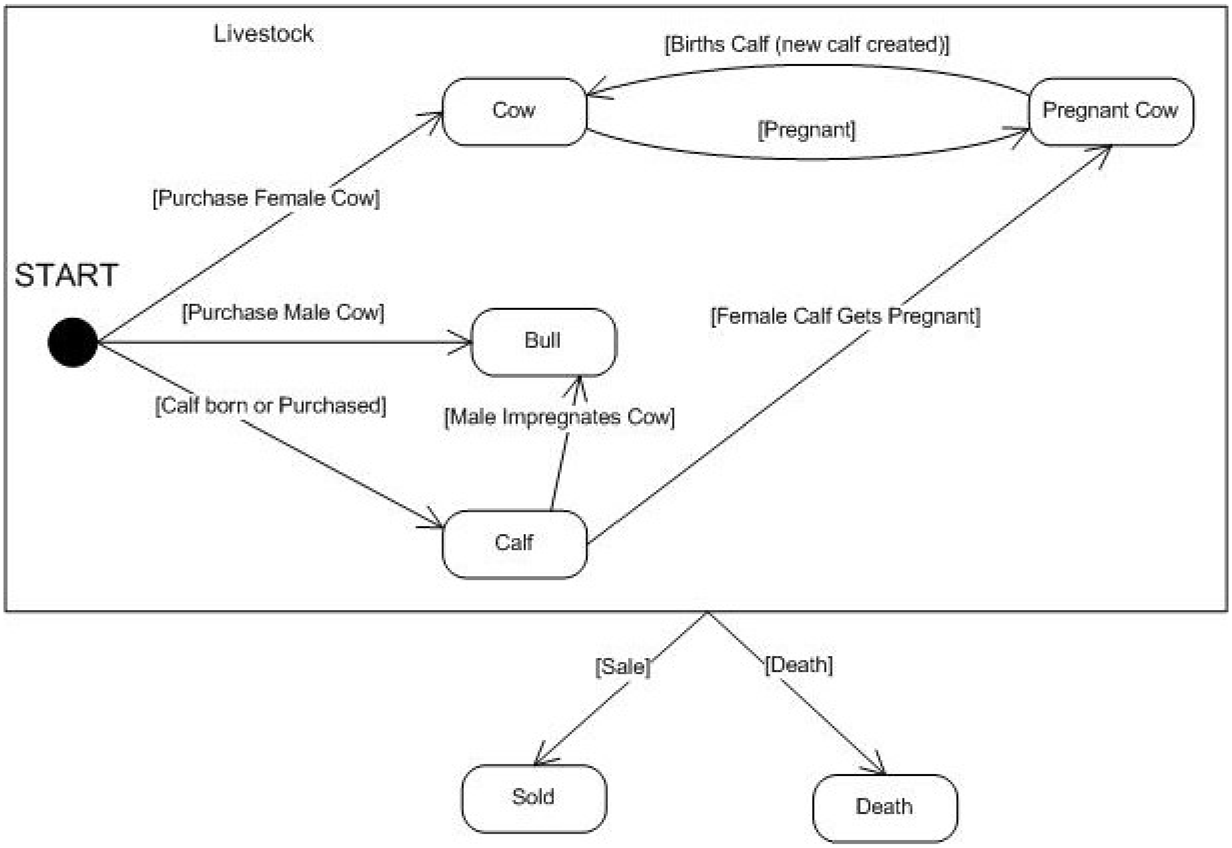


Figure 13: Livestock UML State Diagram

Here we can see that a new animal may be acquired either by purchasing or by being born to a cow on the farm. In both instances, data on the new animal should be collected. If a calf is retained as a replacement cow or bull, the transition of this calf to a mature animal is also important because it changes the purpose of the animal from a potential product to a producer. When a livestock is sold, the event must be captured because the animal no longer exists on the farm. Likewise, the death of an animal will remove it from the farm. Depending on the cause of death, the farmer may have to take further actions which may affect the economic analysis because a potential producer or product was lost.

Figure 14: Birth Event User Form

Figure 15: Feed Purchase Event User Form

Other key events dealing with the management of livestock include feeding, medicating, weaning, and changes in an animal's herd. In each of these instances, the user has the ability to enter relevant information about the event. The more data that is entered, the better the analysis will be.

The resource management events include purchasing, using, or wasting feed and medication as well as tracking any additional costs. This portion of the system includes managing the feed and medication inventories as well by automatically adjusting the inventory when animals are fed or purchases are made.

We focused on the key events of the farm because these are the instances when farmers are likely to take notes. We also aimed to give the user degrees of freedom in the specific format of the information that they enter. This is a tradeoff; the more specific information required, the more analysis available, but it was also important that the application fit well into how farmers do their work.

UI Guidelines

Because the nature of the cow-calf operation requires farmers to be out in the field surrounded by large animals, it is important that data collection with this system be easy and intuitive so attention may be given to the job at hand. Because the typical user interaction methods of a PDA make it difficult to enter specific values, we worked to reduce the amount of data the user actually has to enter by replacing, where possible, text entries with lists of choices. In situations where more freedom is required, we offer users the option to both pick a common value from a list or to enter their own. This enables users to enter data more quickly and more consistently, making the data collection process more effective and easier for the user.

Navigation is a big part of system usability. This is especially true for a PDA application because the limited viewing area potentially increases the total number of forms. This makes the screen flow very important. Our design offers a logical flow, always offering links to the forms we believe will be commonly viewed.

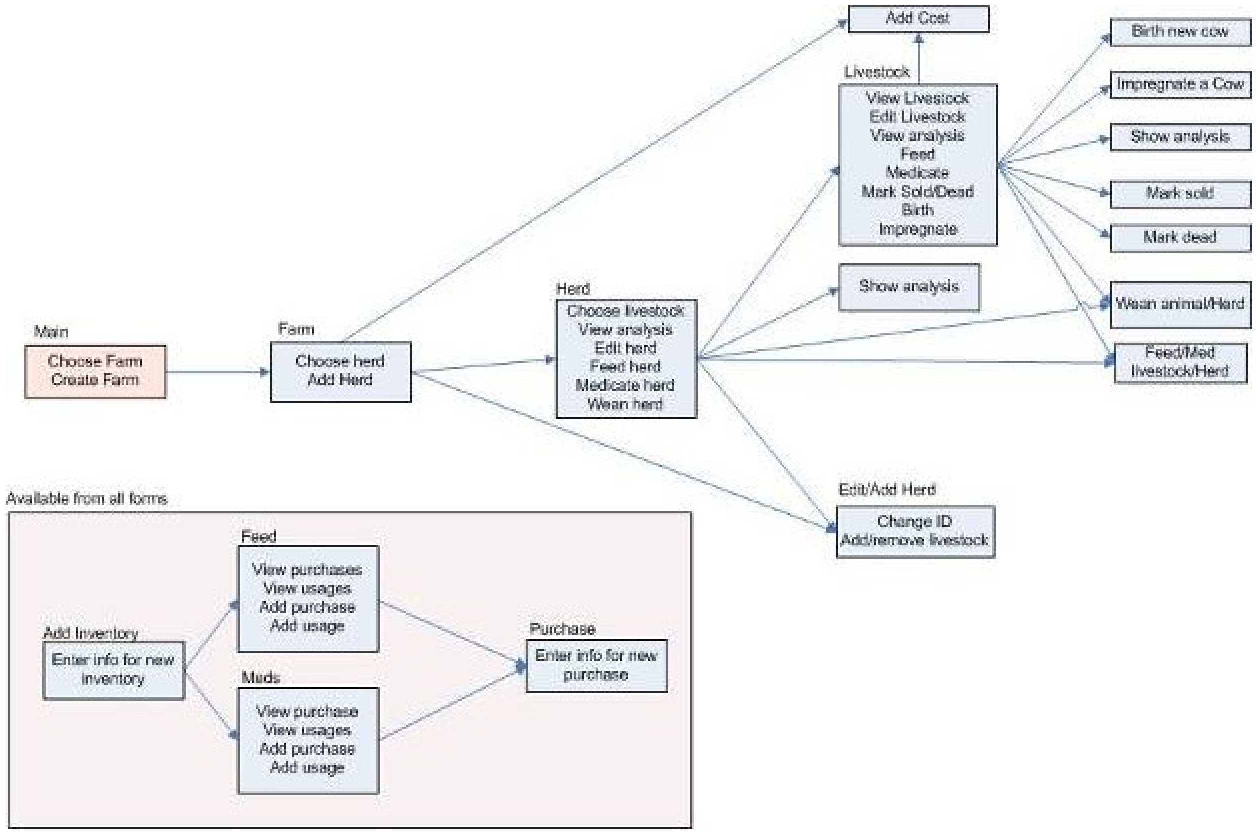


Figure 16: UI Screen Flow

The point made earlier about the hierarchical design of the farm, herds, and livestock is more clear here. The user must traverse through this hierarchy to get to any animal. This allows us to offer a consistent model for finding data and keeping the user from becoming overwhelmed with too much information.

V. CONCLUSION

Validation Plan

Validation of this system can be broken into two basic categories: user acceptance and testing the management usefulness of the software. User acceptance could be tested by creating focus groups composed of farmers. Each person should be timed while performing a number of tasks, such as creating a herd or purchasing grain. Additionally, a survey focusing on system usability should be administered to gather specific information about what tasks were especially difficult or easy. There are existing questionnaires available that measure the quality of software from the user's perspective that suit our needs. One such survey is the Software Usability Measurement Inventory (SUMI) that promises reliable results with as few as 12 users.

To testing the usefulness of this system, a more in depth study would be necessary. The overall goal of the system was to improve profitability of a small to medium sized farm with the help of technology as a tool for real-time data collection and analysis. To accurately asses this, it would be necessary for the system to be in use for an entire farming cycle which is from weaning to weaning. That requires a full year of use, as well as access to the farm's past financial status so profitability can be measured relative to that specific farm.

For the full experiment, it would be necessary to measure the effectiveness of at least three farms over a full year, with bi-monthly update meetings with each farmer to document the process. One farm would use the new system, another would use a paper and pencil method, and a third would collect no data. After each meeting, the finances of each farm could be compared to each other and the farm's past records to search for any correlations. More comprehensive study might include testing this system against a post-processing system such as CHAPS, but for an initial study, this is not necessary.

Future Work

There are multiple opportunities for future work. Below we highlight a few independent areas of work.

In order to allow this application to scale to larger operations, it is necessary to deal with the possibility of multiple people using the system concurrently; that is, two or more people using separate PDAs to collect data. In this situation, it is important that all users' changes be recorded and saved. This will require the users' data to be merged with no data being overwritten. To do this there are two approaches, one is to complete the merge at the end of the day on the PC. For this option, there are numerous existing applications for merging files, and some even specifically for tree structured data like XML, most notably, the 3DM application discussed in [Lindholm 2004]. However, there could be benefits to allowing individual PDAs to merge their current data. This option requires more work because merging is an expensive operation and an application for the PDA may not be readily available.

In addition to handling multiple users, a PC companion for this application would have numerous benefits. This would offer a much more simple method for entering longer data inputs, such as the Animal Identification Number. A PC application would also offer the ability to perform additional and more comprehensive analysis on the collected data. Because the data is in a standard XML format, it would be easy to integrate any available tools for data analysis.

Because this system was built with extensibility in mind, it would be useful to put this architecture to use and create a family of applications for the domain of farming. Going through the process of modifying this application to track a different type of operation would identify any holes in encapsulation and data hiding in the design. Once a generalized framework is identified, it could be used to allow developers to create customized applications.

In addition to using technology as a method of data collection and analysis it would also be interesting to apply an entirely new style of management. The Balanced Scorecard [Kaplan et al. 1992] is a management system that focuses on the balance between different aspects of a business by gathering metrics that help quantify the success of the business. The Balanced Scorecard works by breaking a business into four separate views: financial, customer, internal business process, and learning and growth; each with objectives and measures that reflect the business' overall vision and strategy. These four elements of a company often have conflicting interests and if managed separately, sub-optimization will occur. For example, a goal of a shortened response time, will conflict with a promise to become more customer oriented.

This style of management provides feedback from each business perspective which allows managers to consistently stay in the loop, giving them a good idea of how efficient and productive their processes are working. Because of this constant process feedback, it is easier to make continuous improvements throughout the different processes thereby stopping problems before they occur.

One of the most difficult issues with the Balanced Scorecard is coming up with good metrics for each section. Much of the decision of what to measure is based on the individual company's priorities, but good metrics tend to share similar qualities, such as giving decision makers a representative status of the organization, process diagnostic feedback for improvement, and showing useful performance trends over time. This management style has been applied to many IT organizations but it would be interesting to see its affects on farming.

To conclude, this project has provided a solid code base. From here, research can direct this project in multiple directions dealing with many different areas of both computing and agriculture.

Literature Reviewed

Industry Information & Resources. http://www.ohiobeef.org/industryinfo.htm.

Animal Identification: NAIS Key Components. http://animalid.aphis.usda.gov/nais/subjects/animal_id/index.shtml.

[Arveson 1998]
Arveson, Paul, (1998). What is the Balanced Scorecard?, http://www.balancedscorecard.org/basics/bsc1.html.

[Faulk 1997]
Faulk, S. R. 1997. Software Requirements: A Tutorial. In *Software Engineering*, M. Dorfman and R. H. Thayer eds., IEEE Computer Society Press (1997), 82-103.

[Hobart 2001]
Hobart, James (March 2001). Designing Successful Mobile Applications. http://www.classicsys.com/css06/cfm/article_2001_03.cfm

[Hobart 1995]
Hobart, James, (October 1995). *Principles of Good GUI Design*. http://www.classicsys.com/css06/cfm/article_1995_10.cfm

[Hughes et al. 1989]
Hughes, Harlan, Toman, Norman, Aakre, Dwight, Boyles, Stephen, (1989). *Preparing and Understanding a Beef Cow-Calf Enterprise Budget.* North Dakota: NDSU Extension Service.

[IRM]
Integrated Resource Management. (2001). Lewiston, Maine: Geiger.

[Kaplan et al. 1992]
Kaplan, Robert S. & Norton, David P. (1992). The Balanced Scorecard- Measures That Drive Performance. *Harvard Business Review*, January-February 1992, 71-79.

[Lindholm 2004]
Lindholm, Tancred, (2004). A Three-way Merge for XML Documents. *Proceedings of the ACM Symposium on Document Engineering*, October 2004.

[Mangione]
Mangione, David A *Scoring Cows Can Improve Profits.* http://ohioline.osu.edu/l292/index.html

[Miller et al. 2001]

Miller, A. J., Faulkner, D. B., Pas, Knipe, R. K., Strohbehn, D. R., Parrett, D. F. and Berger, L. L., (2001). Critical Control Points for Profitability in the Cow-Calf Enterprise. *The Professional Animal Scientist, Issue 17.* 295-302.

[Parnas 1972]
Parnas, D.L., (1972). On the Criteria to be Used in Decomposing Systems into Modules. Communications of the ACM, Vol. 15, No. 12. 1053-1058.

[Pascoe et al. 2000]
Pascoe, Jason, Ryan, Nick, & Morse David, (2000). Using While Moving: HCI Issues in Fieldwork Environments. *ACM Transactions on Computer-Human Interaction, Vol. 7, No. 3*, 417-437

[Schraefel et al. 2004]
Schraefel, m.c., Hughes, Gareth V., Mills, Hugo R., Smith, Graham, Payne, Terry R. & Frey, Jeremy, (2004). Breaking the Book: Translating the Chemistry Lab Book into a Pervasive Computing Lab Environment. *CHI Letters Volume 6, Number 1*, 25-32.